WS-I Basic Security Profile Enhanced Logging Specification Requirements

Document Status: Working Group Draft
Version: 1.0
Date: 8 May, 2006
Editor: Ram Poornalingam, Microsoft Corporation
Contributors: Craig Chaney, IBM
Keith Stobie, Microsoft

Revision History

July 25, 2005 Initial version
May 8, 2006 Working Group Draft

Copyright

1. Introduction

Verifying Basic Security Profile conformance requires SOAP stack instrumentation. This specification addresses why instrumentation is necessary and how it can be achieved. This document assumes that the reader understands the usage of the Interoperability testing tools version 2.0 [1].

Complete BSP verification of encrypted SOAP message emitted by the application is not possible. The reason being, Basic Profile verification, a requirement of the BSP, of encrypted messages is not possible. To achieve BP verification, the unencrypted form of the message is necessary. The profile conformance coverage that can be achieved, without adhering to this specification, is only at the surface level.

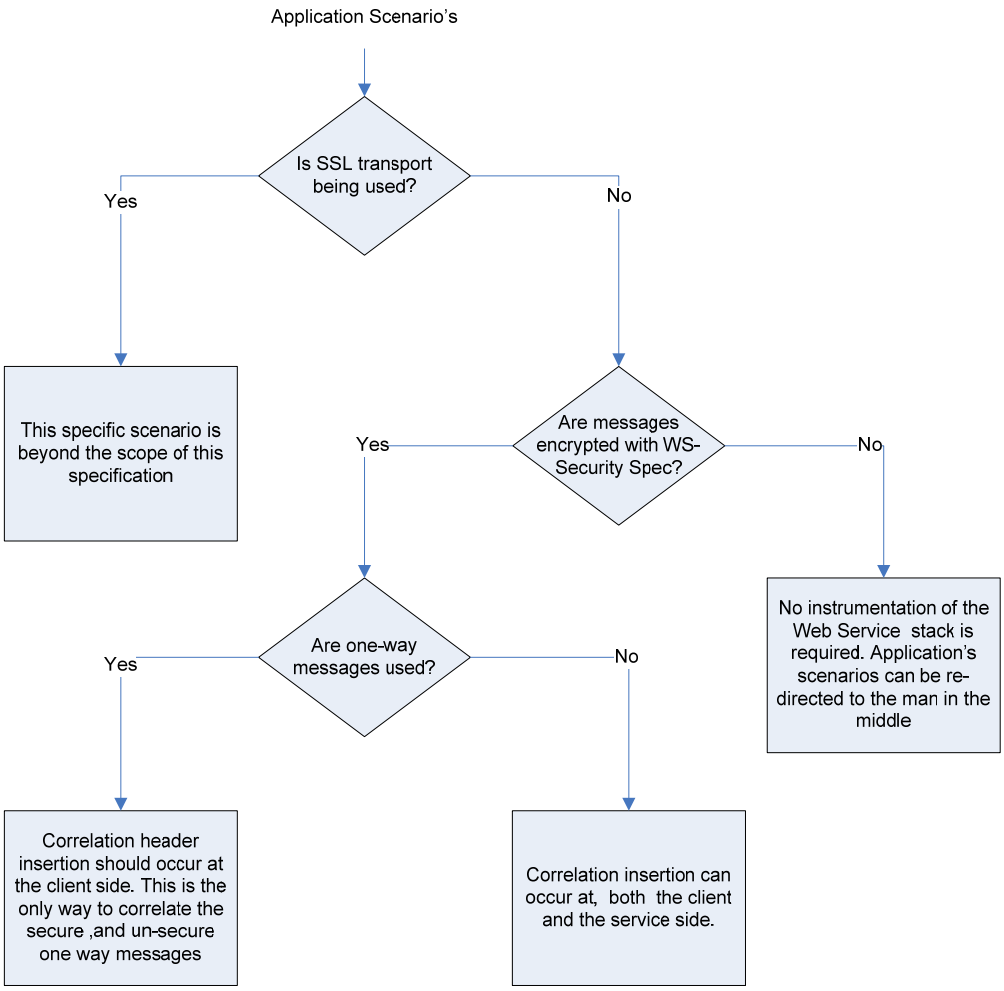
2. How to Use this Document

This specification is primarily aimed at web services developers. The web services developer may not need to instrument the web service stack; since a plug-in may be provided by the web service stack vendor to do the same.

3. Usage Scenarios

3.1. Scenarios

The decision tree below describes the various options.



SSL scenario is out scope for this specification. Apart from the SSL scenario, there are broadly two mainline scenarios. Scenario one, the non encryption scenario – in which case the WS-I tools can be used without a need for a stack specific plug-in. Scenario two, the encryption scenario – where there is a need for stack specific plug-in. The last decision box in the above diagram illustrates that, for 1-way messages correlation header insertion has to always take place at the client side.

3.2. Usage Steps

Step 1. Install and setup the toolset

The Interoperability toolset version 2.0[1] needs to be installed.

Step 2. Instrument the SOAP stack

Instrumentation can be done using a plug-in mechanism which can be re-used across applications. This plug-in maybe provided by the SOAP stack vendor; if not, then the web service developer is needs to provide it.

Step 3. Execute the Application Scenarios

The application scenarios are executed over the instrumented SOAP stack.

Step 4. Collect and Merge the logs

Once the log has been collected at the endpoint, it is merged with the log file collected by the Monitor tool. This merge is done through the Log Merge Tool provided along with the Interoperability Testing Toolset [1]. Correlation can be performed based on other headers like WS-Addressing MessageId header. In order to override the standard behavior, the correlation header details need to be provided to the log merge tool

Step 5. Run merged log through the Analyzer

The merged log is then supplied to the Analyzer tool provided with the Interoperability tool to do BSP verification.

Note the use of the Log File format may be considered a covert channel and has privacy issues. Because the envelope that will be given to the WS-I Test Tool for logging may include **Personal Identifiable Information** (PII), it is imperative that those providing the ENVELOPE to the test tools mask or remove the PII. The message may have had encryption applied for a good reason. The Test Tools are only looking inside well known elements that shouldn't have PII or only verifying existence of elements (against WSDL) for elements that may have PII. It is beyond the scope of the test tools to remove the PII. Users of the test tools when logging information should make sure they are either working with dummy or test data with no PII or else mask or remove the PII data.

4. Correlation Mechanism

4.1. Algorithm

In order to do correlation between the encrypted and unencrypted forms of the message, a Correlation header is necessary. This specification provides one such header. It is optional to use this header. This header should not be encrypted. Also, the unencrypted form of these messages needs to be logged at the endpoints. The table below illustrates the header insertion and logging mechanism. The client and server side insertion is used when the SOAP stack cannot correlation the request and replies messages.

Scenarios	Client and Service Side header Insertion	Client Side Insertion	Service Side Insertion
Client Side Logging	• **The client inserts the correlation header and logs the messages on outgoing request** • **Service on reply (on a RR pattern) either can insert the same correlation**	• **The client inserts the correlation header.** • **On the correlated reply from service the client re-inserts the same correlation. The client then logs the message.**	**NA**

	header or insert a new header with new id • **The incoming reply is logged by the client**	• **It is the client's responsibility to correlate the request and reply.**	
Service Side Logging	• **The client inserts the correlation header.** • **The service on incoming request logs the messages** • **Service on reply (on a RR pattern) can either insert the same header or insert a new header with new id.** • **The outgoing reply is logged by the service**	**NA**	• **The service on incoming request inserts the correlation header and logs the message.** • **The service has to correlate the request and reply, and insert the same correlation header on the outgoing reply as it did on the request. The service then logs the message.**

Table 1: Correlation and Logging Scenarios

5. Samples

5.1. Sample SOAP Message with Correlation Header

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Header>
      <wsicor:Correlation wsicor:CorrelationId="UUID:2dd21cd9-f713-
471f-9fd0-cb28a21c667d" xmlns="http://ws-
i.org/testing/2005/04/correlation" xmlns:wsicor="http://ws-
i.org/testing/2005/04/correlation">
        <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
          <u:Timestamp u:Id="_0" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
            <u:Created>2005-06-02T22:39:05.902Z</u:Created>
            <u:Expires>2005-06-02T22:54:05.902Z</u:Expires>
          </u:Timestamp>
          <e:EncryptedKey Id="uuid-3fb9effc-b1d1-4b42-86c1-
56d76aea5e3e-1" xmlns:e="http://www.w3.org/2001/04/xmlenc#">
            <e:EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-
mgf1p"></e:EncryptionMethod>
            <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
              <o:SecurityTokenReference>
                <X509Data>
                  <X509IssuerSerial>
                    <X509IssuerName>CN=Root Agency</X509IssuerName>

<X509SerialNumber>242965E03B49B58749F1E953894BD34E</X509SerialNumber>
                  </X509IssuerSerial>
```

```
                </X509Data>
              </o:SecurityTokenReference>
            </KeyInfo>
            <e:CipherData>…….
```

The sample illustrates a message with the WSI correlation header. The message needs a header for correlation, but the use of WS-I correlation header is optional.

5.2. Sample Correlation Log

```
<?xml version="1.0" encoding="utf-8"?>
<log xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.ws-
i.org/testing/2005/04/correlationlog/">
  <configuration>
    <log>client</log>
  </configuration>
  <messageEntry correlationId="UUID:233d1faf-b531-4ad1-bb7f-
acdd9c08153e" messageType="request">
    <messageContent>&lt;s:Envelope
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"&gt;&lt;s:Header&gt;
&lt;a:Action
s:mustUnderstand="1"&gt;http://Microsoft.ServiceModel.Samples/ICalculat
or/Add&lt;/a:Action&gt;&lt;a:MessageID&gt;uuid:c1fa673a-6cd4-4f31-86b4-
24de3ba96229&lt;/a:MessageID&gt;&lt;Correlation
wsicor:CorrelationId="233d1faf-b531-4ad1-bb7f-acdd9c08153e"
xmlns:wsicor="http://ws-i.org/testing/2005/04/correlation"
xmlns="http://ws-i.org/testing/2005/04/correlation"
/&gt;&lt;/s:Header&gt;&lt;s:Body&gt;&lt;Add
xmlns="http://Microsoft.ServiceModel.Samples"&gt;&lt;n1&gt;100&lt;/n1&g
t;&lt;n2&gt;15.99&lt;/n2&gt;&lt;/Add&gt;&lt;/s:Body&gt;&lt;/s:Envelope&
gt;</messageContent>
  </messageEntry>
  <messageEntry correlationId="UUID:233d1faf-b531-4ad1-bb7f-
acdd9c08153e" messageType="response">
    <messageContent>&lt;s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"&gt;&lt;s:Header&gt;
&lt;Correlation wsicor:CorrelationId="233d1faf-b531-4ad1-bb7f-
acdd9c08153e" xmlns=http://ws-i.org/testing/2005/04/correlation…….
&lt;s:Body u:Id="1" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd"&gt;&lt;AddResponse
xmlns="http://Microsoft.ServiceModel.Samples"&gt;&lt;AddResult&gt;115.9
9&lt;/AddResult&gt;&lt;/AddResponse&gt;&lt;/s:Body&gt;&lt;/s:Envelope&g
t;
</messageContent>
 </messageEntry>
</log>
```

The above example is of client side logging with only client side insertion. Notice that the correlationId of both the request and response messages are the same.

6. Schemas

6.1. Correlation Header Schema

```
<xs:schema     xmlns:tns=http://ws-i.org/testing/2005/04/correlation
elementFormDefault="qualified"  targetNamespace="http://ws-
   i.org/testing/2005/04/correlation"    xmlns:xs=
   "http://www.w3.org/2001/XMLSchema" >
  <xs:element name="CorrelationHeader" nillable="true"
   type="tns:CorrelationHeader" />
  <xs:complexType name="CorrelationHeader">
    <xs:attribute name="correlationId" type="xs:anyURI" />
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
</xs:schema>
```

It is optional to follow the above schema. Any correlation header mechanism can be used for example WS-Addressing.

6.2. Correlation Log (Client/Server) Schema

```
<xs:schema xmlns:tns="http://www.ws-
i.org/testing/2005/04/correlationlog/" xmlns:wsi-monlog="http://www.ws-
i.org/testing/2005/01/log/"  elementFormDefault="qualified"
targetNamespace="http://www.ws-i.org/testing/2005/04/correlationlog/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://www.ws-i.org/testing/2005/01/log/"
schemaLocation="log.xsd"/>
  <xs:element name="log" nillable="true" type="tns:log" />
  <xs:complexType name="log">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="configuration"
type="tns:configuration" />
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="messageEntry" type="tns:messageEntry">
        <xs:unique name="ID_PK">
          <xs:selector xpath="messageEntry" />
          <xs:field xpath="@correlationId" />
        </xs:unique>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="timestamp" type="xs:dateTime" />
  </xs:complexType>
  <xs:complexType name="configuration">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="implementer"
type="wsi-monlog:implementation" />
      <xs:element minOccurs="1" maxOccurs="1" name="log"
type="tns:logType" />
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" use="optional" />
```

```
  </xs:complexType>
  <xs:simpleType name="logType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="client" />
      <xs:enumeration value="service" />
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="messageEntry">
    <xs:complexContent>
      <xs:extension base="wsi-monlog:messageEntryBase">
        <xs:attribute name="messageType" type="wsi-
monlog:rrMessageType" use="required" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

It is necessary to adhere to the logging schema.

7. References

[1] See the Java or C# implementation of the Interoperability Testing Tools, available from http://www.ws-i.org/deliverables/workinggroup.aspx?wg=testingtools (Version 2.0 not yet available at time of publication.)